AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A multithreaded processor supporting a plurality of active threads, the multithreaded processor comprising:

an instruction fetch and issue unit comprising:

an instruction fetch stage configured to fetch a plurality of sets of ~~fetched~~ bits, wherein each set of ~~fetched~~ bits ~~represents~~ can represent a plurality of~~one or more~~ instructions; and

a pipeline coupled to the instruction fetch ~~and issue unit~~stage and configured to receive a set of fetched bits and an associated thread ID.

2. (Original) The multithreaded processor of claim 1, wherein the instruction fetch and issue unit further comprises an instruction buffer coupled to the instruction fetch stage and configured to store the sets of fetched bits and the associated thread ID for each set of fetched bits.

3. (Original) The multithreaded processor of claim 1, wherein the instruction fetch and issue unit further comprises a pre-decode stage configured to pre decode each set of fetched bits.

4. (Original) The multithreaded processor of claim 1, wherein the pipeline further comprises a plurality of pipeline stages, wherein each pipeline stage stores a thread ID.

5. (Currently Amended) The multithreaded processor of claim 3, wherein the pipeline further comprises a data forwarding unit for forwarding data from a first pipeline stage having a first thread ID to ~~an~~ a second pipeline stage having a second thread ID.

6. (Original) The multithreaded processor of claim 5, wherein the data forwarding unit comprises a thread ID comparator.

7. (Original) The multithreaded processor of claim 5, wherein the data forwarding unit forwards data when the first thread ID is equal to the second thread ID.

8. (Original) The multithreaded processor of claim 5, wherein the data forwarding unit prevents data forwarding when the first thread ID is not equal to the second thread ID.

9. (Original) The multithreaded processor of claim 1, further comprising a trap handler.

10. (Original) The multithreaded processor of claim 9, wherein the trap handler resolves a first trap having a first thread ID when the active thread corresponds to the first thread ID.

11. (Original) The multithreaded processor of claim 9, wherein the trap handler suspends a first trap having a first thread ID when the active thread does not correspond to the first thread ID.

12. (Currently Amended) A method of operating a multithreaded processor supporting a plurality of threads, the method comprising:

fetching a first set of ~~fetched~~ bits, ~~representing one or more~~which can represent a plurality of instructions;

attaching an associated thread ID to the set of fetched bits; and

issuing the instructions of the first set of fetched bits with the associated thread ID to a pipeline.

13. (Original) The method of claim 12, further comprising:

reading a first operand for the first set of instructions; and

propagating the associated thread ID through the pipeline with the first set of instructions and operand.

14. (Original) The method of claim 12, further comprising storing the associated thread ID in a thread ID memory in each stage of the pipeline operating on an instruction from the first set of instructions.

15. (Original) The method of claim 12, further comprising forwarding data from a first pipeline stage to a second pipeline stage when a first thread ID in the first pipeline stage is equal to a second thread ID in the second pipeline stage.

16. (Original) The method of claim 12, further comprising preventing data forwarding from a first pipeline stage to a second pipeline stage when a first thread ID in the first pipeline stage is not equal to a second thread ID in the second pipeline stage.

17. (Original) The method of claim 12, further comprising storing an active thread ID of an active thread as a trap thread ID when a trap is detected.

18. (Original) The method of claim 17, further comprising preventing trap resolution when the trap thread ID does not equal the active thread ID.

19. (Original) The method of claim 17, further comprising resolving a trap when the trap thread ID equals the active thread ID.

20. (Currently Amended) A multithreaded processor supporting a plurality of threads comprising:

means for fetching a first set of ~~fetched bits representing one or more~~, which can represent a plurality of instructions;

means for attaching an associated thread ID to the set of fetched bits; and

means for issuing the instructions of the first set of fetched bits with the associated thread ID to a pipeline.

21. (Original) The multithreaded processor of claim 20, further comprising:

means for reading a first operand for the first set of instructions; and

means for propagating the associated thread ID through the pipeline with the first set of instructions and operand.

22. (Original) The multithreaded processor of claim 20, further comprising means for storing the associated thread ID in a thread ID memory in each stage of the pipeline operating on an instruction from the first set of instructions.

23. (Original) The multithreaded processor of claim 20, further comprising means for forwarding data from a first pipeline stage to a second pipeline stage when a first thread ID in the first pipeline stage is equal to a second thread ID in the second pipeline stage.

24. (Original) The multithreaded processor of claim 20, further comprising means for

preventing data forwarding from a first pipeline stage to a second pipeline stage when a first thread ID in the first pipeline stage is not equal to a second thread ID in the second pipeline stage.

25. (Original) The multithreaded processor of claim 20, further comprising means for storing an active thread ID of an active thread as a trap thread ID when a trap is detected.

26. (Original) The multithreaded processor of claim 25, further comprising means for preventing trap resolution when the trap thread ID does not equal the active thread ID.

27. (Original) The multithreaded processor of claim 25, further comprising means for resolving a trap when the trap thread ID equals the active thread ID.

28. (New) The multithreaded processor of claim 1, further comprising a trace unit coupled to the instruction fetch and issue unit.

29. (New) The multithreaded processor of claim 28, wherein the trace unit comprises:

a trace generation unit, which monitors the set of fetched bits and the associated thread ID to detect branches, jumps, or calls, and generates a program trace; and

a trace compression unit, which compresses the program trace.

30. (New) The method of claim 12, further comprising monitoring the set of fetched bits and the associated thread ID to detect branches, jumps, or calls, and generates a program trace; and

compressing the program trace.

31. (New) The multithreaded processor of claim 20, further comprising a trace unit coupled to the instruction fetch and issue unit.

32. (New) The multithreaded processor of claim 31, wherein the trace unit comprises:

a trace generation means for monitoring the set of fetched bits and the associated thread ID to detect branches, jumps, or calls, and for generating a program trace; and

a trace compression means for compressing the program trace.